计算机科学 1997 Vol. 24 №.1

一种面向图形化编程的软件设计方法

刘 昱 王立福
(北京大学计算机科学技术系 北京 100871)

摘 要 This article first discusses the inconsistency between traditional software design methods and textual programming languages, then introduces the graphical programming language. With the analysis of its programming properties, the article proposes a software design method for graphical programming paradigm. It not only solves the former inconsistency, but is also proven quite effective.

关键词 Software development process, Software design method, Graphical programming language, Encapsulation, Modularity, Hierarchy.

1 引言

软件开发过程实质上就是从问题域到解空间的求解过程，与一般的问题求解不同，软件开发的问题域是由自然语言描述的现实世界的一部分，而解空间则对应着由程序代码表示的计算机应用。由于两者所对应的领域及所采用的表达手段不同，使软件开发比一般的数学问题求解更为复杂。

长期以来，对软件开发的分析、设计及编码阶段一般采用不同的方法与表示手段，造成软件开发过程中信息不一致，不能充分利用各个阶段的结果，直接影响了软件的质量及开发周期。比如：人们习惯于使用的图形化分析设计方法，如 JSD，Yourdon DFD 图等，旨在记录或描述软件的整体结构，使设计者对软件功能有一个抽象而全面的认识。然而，图形化分析设计方法常常采用流图、框图、表格、文字等方式，缺乏语法及语义定义，属于非形式化方法，与具有精确语法及语义定义的编程语言很难对应起来，主要因为编程语言（指文本语言）具有"1 维"特性，由一系列字符组成，而分析设计结果是图形化的，具有"2 维"特性。

显然，解决这一矛盾有两类方法，一是将分析设计的图形化表示转换成线性表示，使设计方法形式化；二是使编程语言具有"2 维"特性，简化与设计结果之间的对应。实际上，图形化编程语言就采用了这种编程方式。本文将主要讨论图形化编程语言的本质特性，提出一种适用于图形化编程方式的分析设计方法，并举例证明其有效性。

2 图形化编程语言的特性

2.1 图形化编程概述

图形化编程产生于八十年代中期且发展十分迅速，到最近几年已经有了广泛的应用。图形化程序中的代码由图标（icon）及其间连线构成，图标相当于文本语言里的函数模块，图标间的连线完成数据的相互传递。图形化程序好象一张由图标和连线组成的流图或网络图，而不再是一个顺序的指令文件，因此具有二维特性。

图形化编程代表了软件开发的一种新方式。它通过图标来构成代码，编程方式直观，不仅减少了开发时间，而且易于修改，降低开发成本。下面以美国 National Instruments 公司研制的 LabVIEW 环境为例，通过在标准接口仪器自动测试领域的应用，分析图形化语言的编程范型。

2.2 图形化语言的编程范型

LabVIEW 图形编程语言中的基本编程单位是一个 VI（Virtual Instrument），由三部分组成，即面板（Front Panel）、框图（Block Diagram）和连接标志（Connector Patterns）

面板是用户程序的主要输入输出界面。用户通过 Controls 菜单在面板上选择控制及显示机制，完成测试条件的设置及测试结果的显示。框图是测试人员根据测试方案及测试步骤进行测试编程的界面。用户可以通过 Functions 选项选择各种图标组成相应的测试逻辑。连接标志是 VI 的图标表示，包含了有关的输入/输出信息，比如 VI 的名称，其 I/O

参数的个数、类型等。用户在选择 VI 时，不仅需要明确 VI 的功能，而且必须清楚其接口形式，进而通过调用该 VI 的连接标志将其连入程序。连接标志的定义也在面板上完成。

由此可见，在采用 LabVIEW 进行应用软件开发时包括两方面的工作：一方面是句法的定义过程，即选择与布局用户界面上的各类图形对象，诸如按钮、滑块、图表等，故可称为定义界面(Form)；另一方面是语义的定义过程，即通过图形化的程序，将界面对象的动作与其逻辑上应当完成的任务相关联，这时的图形化程序就称为 Formula。这种 Form/Formula 的编程范型将用户界面开发与图形化编程集成起来，其中的每一个 Form(如 LabVIEW 的用户面板)都对应着一个 Formula(LabVIEW 的框图程序)。

2.3 图形化编程的特点

在图形化编程语言中，都存在着一个基本的逻辑编程单位或抽象，如 LabVIEW 中的 VI。这一抽象明确定义了该程序单位与其他类程序单位的边界，强调了它的外部行为(或能够提供的服务)，从而将抽象的外在行为与其内部实现分隔开，因此它具有以下明显特性：

①封装性。封装是对抽象中的各个元素进行分隔的过程，使得抽象的接口与其实现相分离。所谓抽象的接口指抽象的外部视图，它包括该抽象可能为其他抽象提供的服务或要利用其他抽象的哪些操作；抽象的实现即抽象的内部视图，必须完成其接口中规定的全部职责。

②模块化。是指将系统分解成一组高内聚、低耦合的模块。根据开发的需要将关系紧密的抽象组合起来而形成的系统结构就称为模块。使用模块不仅可以降低系统的复杂性，而且有利于软件复用。

③层次化。是对抽象及模块的一种排列，即将它们按某种方式组织起来，便于降低系统的复杂性。在图形化编程中，常见的层次结构反映了抽象间的聚合关系(A is part of B)

④基于数据流。由于图形化程序具有二维特性，其执行过程是基于数据流的，即当且仅当某个模块的入口数据全部到达时，该模块被执行。也正由于这一特性，使得图形化程序的执行具有宏观上的并行性，某一时刻数据可能在多条线路上同时流动，使多个模块的入口条件得到满足，导致多个模块同时被执行。

3 面向图形化编程的软件设计方法——VOBA 方法

3.1 VOBA 方法的特点

在明确了图形化语言的编程范型及特点的基础上，我们提出了一种适用于图形化编程的软件设计方法，即虚拟对象行为分析方法(Virtual Object Behavior Analysis Method)。VOBA 方法是对虚拟对象行为的一种描述，主要用于刻画某一系统的实际运行流程。VOBA 方法方便了由软件设计到图形化程序的转化，有利于设计人员和程序员之间的交流。下面介绍一下该方法的特点：

①VOBA 具有封装性。VOBA 方法的基本逻辑单元是虚拟对象(Virtual Object)。每一个 VO 都对应着图形化程序中的一个 Form/Formula 单元，具有明确定义的接口，能够完成一定的功能，集中体现了方法的封装特性。

VO 的接口(外部视图)说明 VO 所能提供的服务以及它的输入、输出条件，同时还要对所对应的 Form/Formula 单元进行必要的描述，比如该 VO 是否具有 Form，Form 是显式的还是隐式的，Form 中要包含哪些类型的控制及显示机制，对 Formula 的实现有哪些约束等。而 VO 的具体实现(内部视图)由 Formula 的编制者按接口规定完成。

②VOBA 是模块化的。VOBA 方法中的每个 VO 都可以被视为一个模块。它既可以十分简单，由单一的 Form/Formula 单元实现，也可以由多个 VO 相互连接而成，对应多个相互作用的 Form/Formula 单元。VOBA 方法中的模块划分，有利于对程序的理解以及模块的复用。

③VOBA 是层次化的。VOBA 方法能够对系统进行层次化分解，对模块进行排列与组合。这一层次化特性使得该方法能够适用于复杂的大系统开发。

④VOBA 基于数据流。为了描述 VO 之间的数据流关系，VOBA 方法除了使用实线箭头表示数据的流向之外，还引入了交汇点来刻画多条数据流汇合处的数据流向决策。可见，VOBA 的交汇点机制将数据流与控制流结合起来，以适应图形化编程基于数据流的特点。设计中的交汇点往往直接或间接地对应着编程语言中的条件、分支、循环结构，开发人员可以根据经验将它们映射到程序中。另外，VO-

BA 方法还使用虚线箭头及决策单元刻画程序中特殊的控制流向。

可见，从理论上看 VOBA 方法与图形化编程具有相似的特性，适合于对图形化程序进行设计。下面详细介绍 VOBA 方法的基本元素。

3.2 VOBA 方法

VOBA 方法的基本元素包括虚拟对象、连接、交汇点、决策单元和参照表。

3.2.1 虚拟对象(VO) 任何一个软件系统中总是存在着各种类型的对象。它们之间相互连接、相互作用，实现一定的系统功能，实际上，它们是一些结构与行为的封装体，在 VOBA 中用 VO 来表示。

VOBA 流图主要用来描述系统内虚拟对象间的行为关系，其中的对象用 VO 方框表示。当用户认为图中的 VO 信息不够完备时，便可以参照该 VO 的规格说明。当某一个 VO 十分复杂，即使用说明文件也不能描述清楚时，便可以将其分解为若干子 VO，并用 VOBA 流图对这些子 VO 进行描述，以便获得更详细的信息。

3.2.2 连接(Links) VOBA 方法中的连接有两种类型：数据流连接和控制流连接。①数据流连接：用实线箭头表示，它描述系统的各个虚拟对象间数据的流动方向，具有常规的顺序含义。②控制流连接：用虚线箭头表示，它描述系统有关虚拟对象间控制上的关系。它不需要传递数据，具有时间、逻辑、因果等多方面的含义。

3.2.3 交汇点(Junctions) 主要用来对实际流程中的各种分支情况进行逻辑处理。它可以描述一个数据流分解为多个分支的现象，也可以描述多个分支汇合成一个数据流的情况。同时，它还表明了各条分支在逻辑上或时间上的关系。

首先，根据逻辑语义 VOBA 中的交汇点可以分为与(&)、或(O)、异或(X)三种；其次，根据其作用是会聚还是发散(收敛性)，可分为扇入和扇出两类；最后，根据其时间上的协调关系，可以分成同步和异步两种。交汇点的分类情况如图 1，其中的每种组合都

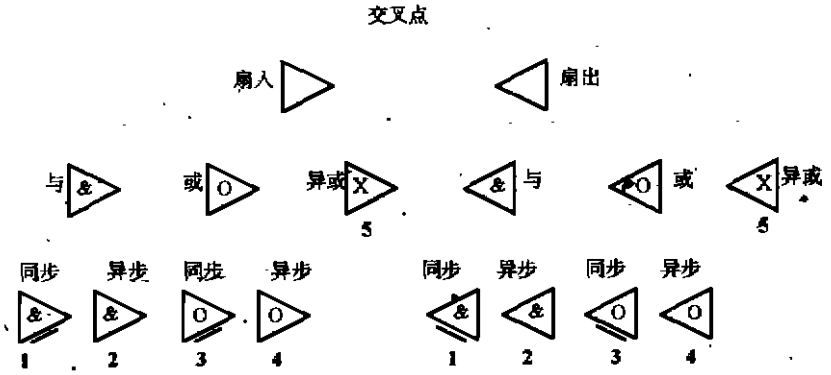


图 1 交叉点的分类及语义

有各自的语义。

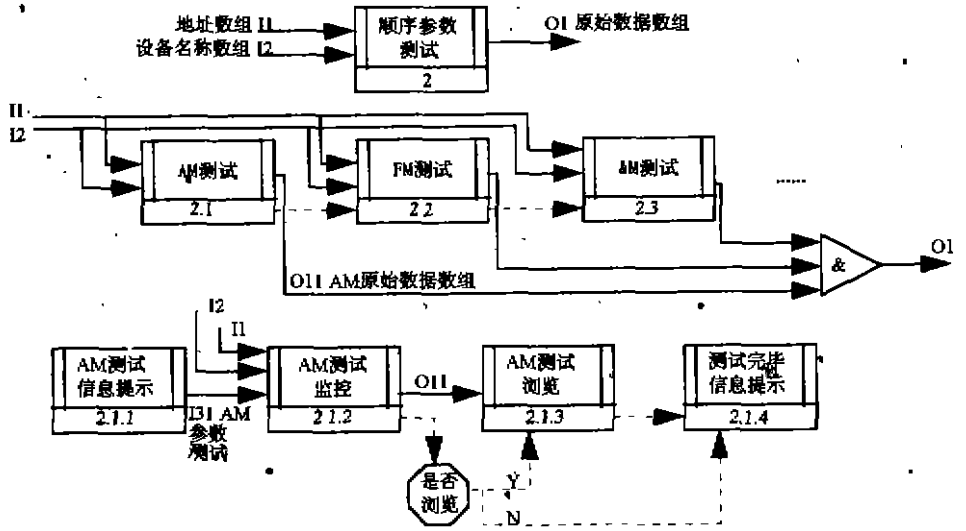
3.2.4 决策单元(Decision Unit) 决策单元对应着程序中的对话框，它将用户的干预引入程序设计，用以改变程序的执行流程。在 VOBA 中，决策单元用六边形表示。由于决策单元的输入、输出信息都属于控制信息，故均由虚线连接。

3.2.5 参照表(Referents) 可以避免图上出现信息聚集的现象，可以对某个 VO、VO 规格说明、交汇点、连接、决策单元等进行参考描述。

3.3 实例

调制度计量测试系统主要完成对调制度分析仪(HP8901)或测试接收机(HP8902)的计量检定工作，并计算出被检仪器的调制精度，对精度在误差限范围之内的仪器给出检定合格证书。本系统的设计

采用了 VOBA 方法，其中包括以下模块：①系统初始化，完成对标准调制信号的标定及系统配置；②顺序参数测试，完成调制度计量测试系统的调幅 AM，调频 FM，调相 &M，AM 解调频响，FM 解调频响等八项参数的测试；③原始数据查询，对本次测量结果进行列表、打印。图 2 给出了顺序参数测试的部分 VOBA 流图及有关的 VO 规格说明。这一测试软件的设计充分利用了 VOBA 方法的前述特点，与 LabVIEW 图形编程语言无缝衔接，大大降低了系统的开发难度及开发成本，同时增强了系统的灵活性。



Name: AM测试监控	No.: 2.1.2
In: I1, I2, I31	Out: O11
Form: Existence(T) Visibility(T)	
Formula: 监控方式以子VO方式开发	
Others: 监控方式多样，如列表、曲线、直方图	

图 2　顺序参数测试的部分 VOBA 流图及有关的 VO 规格说明

4 结 论

图形化程序的二维特性，为保证软件开发中分析、设计、编码各阶段信息的一致性提供了极好的解决契机。除了将图形化分析设计结果线性化、形式化的办法之外，还有一些其他的途径。比如：面向对象的概念与技术就为改进软件开发过程开辟了新的道路。对象不仅提供了对现实世界的某些方面进行描述的方法，同时也可以用于描述软件的实现。因此，很多人都在研究如何运用对象概念描述这两个层次之间，即分析与设计阶段的活动。有关这一部分的内容将另著文章论述。

参考文献

[1] Stephen Paynter, Structuring the semantic definitions of graphical design notations, Soft. Eng., May 1995

[2] Willian S. Bennett, Visualizing software: A graphical notation for analysis, design, snd discussion, 1993

[3] S. W. Clyde et al., Tunable formalism in Object-Oriented analysis: Meeting the needs of both theoreticians and practitioners, OOPSLA'92, 1992

[4] Agentsheets: A medium for creating domain-oriented visual languages, Computer, 28(3), 1995

[5] Why looking isn't always seeing: Readership skills and graphucal programming, CACM, June 1995

[6] Virtual Instrument—When a tool isn't a tool but is, IEEE Potentials, Feb. 1994

[7] 陈禹六等，IDEF0 及 IDEF1X 复杂系统通用的分析设计方法，电子工业出版社，1991